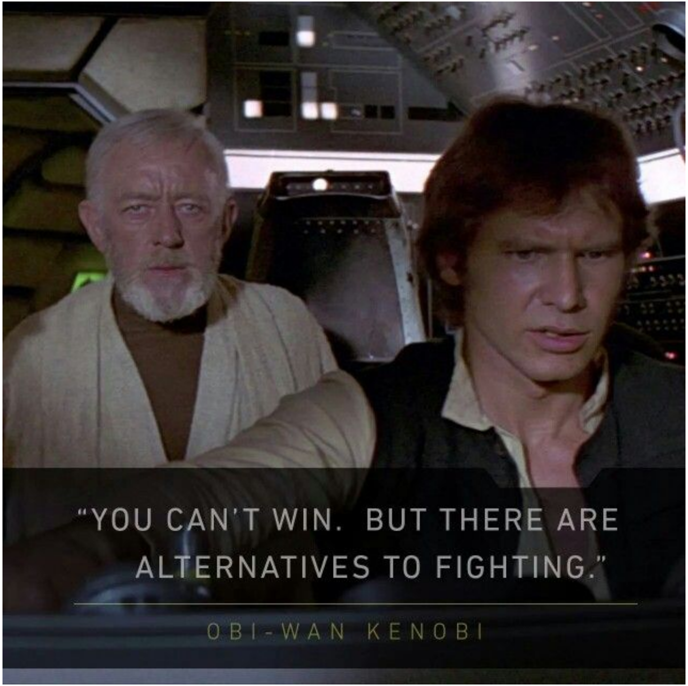
I. Core / C. Advanced analysis techinques

- Limits of comparison-based sorting (**today**)
- Amortized analysis (next week Monday)
- (Begin dynamic programming next week Wednesday)

Today:

- Proof of Theorem 8.1
- Exercises from Section 8.1
- Get head start on amortized analysis



Meme from https://www.pinterest.com/pin/561542647262613858/

You can't comparison-sort in linear time. But there are alternatives to comparisons.

Theorem 8.1. For any comparison-based sorting algorithm, the worst-case number of comparisons is $\Omega(n \lg n)$.

Proof. *For sequences of size n, there are n! permutations, each of which are possible outcomes. Consider the decision tree where each node is a comparison between two array positions.*
Let ℓ be the number of leaves and h the height of the tree. And so

$$\begin{aligned}
n! &\leq \ell && \textit{since every permutation must be a leaf} \\
\ell &\leq 2^h && \textit{since a tree can't have more than } 2^h \textit{ leaves} \\
h &\geq \lg n! \\
&= \Theta(n \lg n) && \textit{by eq 3.19 in CLRS}
\end{aligned}$$

Hence $h = \Omega(n \lg n)$, and thus there must be a permutation reachable by no less than $\Omega(n \lg n)$ comparisons. □

8.1-3.a. Can a comparison-based sorting algorithm have linear running time for at least half the inputs of size n?

Suppose so, that is, suppose there exists a c such that for $\frac{n!}{2}$ of the items, their path is fewer than cn links. This means that in the portion of the tree less than cn links from the root, there are $\frac{n!}{2}$ leaves. In fact, the most possible leaves are 2^{cn}. Thus,

$$\begin{aligned} \frac{n!}{2} &\leq 2^{cn} \\ n! &\leq 2^{cn+1} \end{aligned}$$

$$\begin{aligned} \lg(n!) &\leq cn+1 \\ c &\geq \frac{\lg(n!)}{n} - \frac{1}{n} \end{aligned}$$

Since $\lg(n!) = \Omega(n \lg n)$, there exists a d such that $\lg(n!) \geq dn \lg n$.

$$c \geq \frac{\lg(n!)}{n} - \frac{1}{n} \geq \frac{dn \lg n}{n} - \frac{1}{n} = d \lg n - \frac{1}{n}$$

$\frac{1}{n}$ approaches 0 and $d \lg n$ approaches ∞ (slowly). So, c cannot be a constant. Alternately, let h_1 be the the pseduo-height encompasing the closest $\frac{n!}{2}$ leaves. Observe that $\frac{n!}{2} \leq 2^{h_1}$, and so

$$h_1 \geq \lg n! - 1 = \Omega(n \lg n)$$

8.1-3.b. Can a comparison-based sorting algorithm have linear running time for $\frac{1}{n}$ of the inputs of size n?

Suppose so. Then

$$\frac{n!}{n} \leq 2^{cn}$$

$$\lg(n!) - \lg n \leq cn$$

$$c \geq \frac{\lg(n!) - \lg n}{n} \geq \frac{dn \lg n}{n} \geq d \lg n - \frac{\lg n}{n}$$

Since the $\frac{\lg n}{n}$ term approaches 0, the last expression is increasing. Hence c is not constant.

Alternately, $\frac{n!}{n} \leq 2^{h_2}$, so

$$h_2 \geq \lg n! - \lg n = \Omega(n \lg n)$$

8.1-3.c. Can a comparison-based sorting algorithm have linear running time for $\frac{1}{2^n}$ of the inputs of size n?

Suppose so. Then

$$\begin{aligned}
\frac{n!}{2^n} &\leq 2^{cn} \\
n! &\leq 2^{(c+1)n} \\
\lg(n!) &\leq (c+1)n \\
c &\geq \frac{\lg(n!)}{n} - 1 \\
&\geq \frac{dn \lg n}{n} - 1 \\
&= d \lg n - 1
\end{aligned}$$

Alternately, $\frac{n!}{2^n} \leq 2^{h_3}$, so

$$h_3 \geq \lg n! - n = \Omega(n \lg n)$$

8.1-4. The number of permutations is $\underbrace{k! \cdot k! \ldots k!}_{\frac{n}{k}}$, that is, $(k!)^{\frac{n}{k}}$.

For a decision tree of height h, $(k!)^{\frac{n}{k}} \leq 2^h$. So,

$$\begin{aligned} h &\geq \lg((k!)^{\frac{n}{k}}) \\ &= \frac{n}{k} \lg(k!) \\ &= \frac{n}{k} dk \lg k \quad \text{for some } d \\ &= dn \lg k \end{aligned}$$

Hence $\Omega(n \lg k)$.

For next time

Read Sec 17.(1-3).

Do 17.1-(1 & 3) and 17.2-2.

"Divide and Conquer" problem set due Wed, Sept 25.